**Practical 06**

CODE:

inp\_file = open('src\_code.txt')

macro\_start\_flag = 0

macro\_name\_flag = 0

MDT = dict()

MNT = dict()

ALA = dict()

macro\_name = ''

mdt\_index = 0

mnt\_index = 0

ala\_index = 0

macro\_def = []

macro\_args = []

call\_count = 0

# reading through input and creating MDT table

for line in inp\_file:

if len(line.split()) > 0:

line = line.replace('\n', '')

if line.split()[0] == 'MACRO':

macro\_name\_flag = 0

macro\_start\_flag = 1

pass

elif line.split()[0] == 'MEND':

mdt\_index += 1

macro\_line = (mdt\_index, line)

macro\_def.append(macro\_line)

mdt\_entry = {

macro\_name: macro\_def

}

MDT.update(mdt\_entry)

macro\_def = []

macro\_args = []

macro\_start\_flag = 0

else:

if macro\_start\_flag != 0:

if macro\_name\_flag == 0:

macro\_name = line.split()[0]

mdt\_ent = {

macro\_name: None

}

MDT.update(mdt\_ent)

macro\_name\_flag = 1

mdt\_index += 1

for code in line.split():

# sustituting '#indx' for arguments

if '&' in code:

if code not in [arg[0] for arg in macro\_args]:

arg\_sub = '#'+str(len(macro\_args)+1)

macro\_args.append((code, arg\_sub))

else:

for args in macro\_args:

if args[0] == code:

arg\_sub = args[1]

line = line.replace(code, arg\_sub)

macro\_line = (mdt\_index, line)

macro\_def.append(macro\_line)

else:

# initializing ALA

macro\_ala = []

macro\_call = line.split()[0]

if macro\_call in list(MDT.keys()):

call\_count += 1

arguments = line.split()[1:]

for arg in arguments:

arg = arg.upper()

if len(arg) < 8:

additional\_b = 8 - len(arg)

for b in range(additional\_b):

arg += 'b'

macro\_ala.append(arg)

ala\_entry = {

macro\_call+'\_'+str(call\_count): macro\_ala

}

ALA.update(ala\_entry)

# creating MNT table

for entry in MDT:

mnt\_index += 1

mnt\_entry = {

entry: (mnt\_index, MDT[entry][0][0])

}

MNT.update(mnt\_entry)

# adding values in ALA

ALA\_final = ALA.copy()

for calls in ALA:

macro\_ala = []

macro\_call = calls.split('\_')[0]

given = ALA[calls]

for line in MDT[macro\_call]:

if macro\_call not in line[1] and 'MEND' not in line[1]: for code in line[1].split():

if '#' in code:

index = int(code.replace('#', '')) - 1

macro\_ala.append(given[index])

next\_call = line[1].split()[0]

call\_count += 1

next\_call += '\_' + str(call\_count)

ala\_entry = {

next\_call: macro\_ala

}

ALA\_final.update(ala\_entry)

# printing output

print('\nMacro Definition Table (MDT)')

print('Index\tContents')

for entry in MDT:

for lines in MDT[entry]:

print(lines[0], '\t', lines[1])

print('\nMacro Name Table (MNT)')

print('Index\tMacro Name\tMDT Index')

for entry in MNT:

print(MNT[entry][0], '\t', entry,'\t\t', MNT[entry][1])

print('\nArgument List Array (ALA)')

print('Index\tArgument')

for entry in ALA\_final:

for arg in ALA\_final[entry]:

ala\_index += 1

print(ala\_index, '\t', arg)

**MACROINPUT.TXT:**

MACRO

ADD1 &arg L 1 &arg

A 1 =F'1'

ST 1 &arg

MEND

MACRO

ADDS &arg1 &arg2 &arg3 ADD1 &arg1

ADD1 &arg2

ADD1 &arg3

MEND

ADDS data1 data2 data3 **OUTPUT:**